**Comparing MNB and SVM**

Large Movie Review Dataset

**Maya Mileva**

**HW 7 IST 736**

A picture containing indoor, sitting, shelf, many

Description automatically generated

**Introduction**

You love watching movies. You show up early and breathe in the trailers, mentally noting each coming attraction you would like to see—which is most of them. You invite your friends to linger afterward so you can discuss and digest the film’s best, worst, and most intriguing aspects. If this sounds like you, you might already be a film critic at heart.

A movie review is written with the basic goal of informing the readers about the movie and its concept. Although it seems easy to report all the events happening in the movie and state your opinion, this is a frequent blunder that a lot of people make. While you can express your opinions in a movie review about a film or a documentary, you should do so with a creative and unbiased approach. To a great extent, it depends on the review whether the reader would want to go and see the movie.

A picture containing clock, drawing

Description automatically generated Sentiment analysis is usually used on social media posts and customer reviews in order to automatically understand if some users are positive or negative and why.

So, what exactly is sentiment? Sentiment relates to the meaning of a word or sequence of words and is usually associated with an opinion or emotion. And analysis? Well, this is the process of looking at data and making inferences; in this case, using machine learning to learn and predict whether a movie review is positive or negative.

Maybe you are interested in knowing whether movie reviews are positive or negative, companies use sentiment analysis in a variety of settings, particularly for marketing purposes. Uses include social media monitoring, brand monitoring, customer feedback, customer service and market research (“Sentiment Analysis”).

**Analysis and Models**

**About the data**

The dataset used for this project is IMDB dataset having 50K movie reviews for natural language processing or Text analytics: 25,000 labelled reviews for training and 25,000 reviews for testing.

A drawing of a cartoon character

Description automatically generatedThe dataset can be found here: <http://ai.stanford.edu/~amaas/data/sentiment/>

Due to memory issues only 50% of the data was used. The train and test sets were combined for easier preprocessing. Here are some of the positive and negative reviews:

|  |  |
| --- | --- |
| **Positive** | **Negative** |
| **This unpretentious Horror film is probably destined to become a cult classic. Much much better than 90% of the Scream rip-offs out there! I even hope they come up with a sequel!** | **"ZZZZZZZZZZZZZZZZZZ"! If IMDb would allow one-word reviews, that's what mine would be.** |
| **I just wanted to say that. I love Gheorghe Muresan, so I automatically loved this movie. Everything else about it was so-so... Billy Crystal is a good actor, even if he is annoying. But the thing that made this movie was- at least, for a basketball fan- seeing Gheorghe Muresan act.** | **I never want to see this movie again!<br /><br />Not only is it dreadfully bad, but I can't stand seeing my hero Stan Laurel looking so old and sick.<br /><br />Mostly I can't stand watching this terrible movie!<br /><br />Frankly, there is no reason to watch this awful film. The plot is just plain stupid. The actors that surround Stan Laurel and Oliver Hardy are really really bad and Laurel and Hardy have been funnier in any of their earlier films! <br /><br />I warn you don't watch it, the images will haunt you for a long while to come!** |
| **The story has been told before. A deadly disease is spreading around... But the extra in this film is Peter Weller, his interpretation of Muller on the run is real. He is indeed a desperate person just going home to see his child. This person could be working next to you.** | **This was by far the worst movie I've ever seen. And that’s compared to Alexander, Fortress 2 and The new world.<br /><br />I should go back to blockbuster and ask for my money back along with compensation as it was a truly traumatic experience. For the first ten minutes i was changing the zoom on my widescreen TV because the actors seemed to be out of screen.** |

The dataset looks like this:

A white and black text

Description automatically generated

The data had to be cleaned before modeling. The labels were equally distributed even after getting only 50% of the data(25,000 reviews).

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |

The most common words in the data before cleaning were:

***( 'the', 653644), ('a', 318096), ('and', 315075), ('of', 287032), ('to', 265167), ('is', 205023),***

***('in', 181809), ('i', 149095), ('this', 142985), ('it', 131573), ('that', 130494), ('was', 93322),***

***('as', 89617), ('for', 85187)***

All those stop word have to be removed.

Looking at the word frequency, words like ‘movie’, ‘production’, ‘film/ing’ are dominating.

A picture containing green, sitting, table, hanging

Description automatically generated

Let’s look at the number of words in a sentence next.

A screenshot of a cell phone

Description automatically generated

The maximum length of a sentence is: 394

The average length of a sentence is: 48.54368932038835

Count Vectorizer and Tfidf Vectorizer were used for data preprocessing. Porter Stemmer was added too. Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. For example: words such as “Likes”, ”liked”, ”likely” and ”liking” will be reduced to “like” after stemming. Stemming can result in words that are not actually words. As a result, 4 data frames were created for modeling:

1. *CountVectorizer without stemming*
2. *CountVectorizer with stemming*
3. *Tfidf Vectorizer without stemming*
4. *Tfidf vectorizer with stemming.*

Custom stop words list was added. Max features chosen was 100, due to the really big dataset.

1. *CountVectorizer without stemming*

A picture containing road, photo, meter, black

Description automatically generated

1. *CountVectorizer with stemming*

A picture containing road, black, photo, meter

Description automatically generated

1. *Tfidf Vectorizer without stemming*

A black sign with white text

Description automatically generated

1. *Tfidf Vectorizer with stemming*

*A black sign with white text

Description automatically generated*

**Models**

**MNB**

Multinomial Naive Bayes  is a simple technique for constructing classifiers: models that assign class  labels to problem instances, represented as vectors of feature values, where the class  labels are drawn from some finite set. There is not a single algorithm for training such  classifiers, but a family of algorithms based on a common principle: all naive Bayes  classifiers assume that the value of a particular feature is independent of the value of any  other feature, given the class variable. For example, some fruit may be considered to be an  apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers  each of these features to contribute independently to the probability that this fruit is an  apple, regardless of any possible correlations between the color, roundness, and diameter  features.  For some types of probability models, naive Bayes classifiers can be trained very efficiently  in a supervised learning setting. In many practical applications, parameter estimation for  naive Bayes models uses the method of maximum likelihood; in other words, one can work  with the naive Bayes model without accepting Bayesian probability or using any Bayesian  methods.  Despite their naive design and apparently oversimplified assumptions, naive Bayes  classifiers have worked quite well in many complex real-world situations. In 2004, an  analysis of the Bayesian classification problem showed that there are sound theoretical  reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a  comprehensive comparison with other classification algorithms in 2006 showed that Bayes  classification is outperformed by other approaches, such as boosted trees or random  forests.  An advantage of naive Bayes is that it only requires a small number of training data to  estimate the parameters necessary for classification.

**SVM**

Classifying data is a common task in machine learning. Suppose some given data points  each belong to one of two classes, and the goal is to decide which class a new data point  will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional  vector (a list of p numbers), and we want to know whether we can separate such points  with a (p-1)(p-1)-dimensional hyperplane. This is called a linear classifier. There are many  hyperplanes that might classify the data. One reasonable choice as the best hyperplane is  the one that represents the largest separation, or margin, between the two classes. So, the hyperplane so that the distance from it to the nearest data point on each side is  maximized have to be chosen. If such a hyperplane exists, it is known as the maximum-margin hyperplane  and the linear classifier it defines is known as a maximum-margin classifier; or equivalently,  the perceptron of optimal stability.

More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in  a high- or infinite-dimensional space, which can be used for classification, regression, or  other tasks like outlier’s detection. Intuitively, a good separation is achieved by the  hyperplane that has the largest distance to the nearest training-data point of any class  (so-called functional margin), since in general the larger the margin, the lower the  generalization error of the classifier.

**MNB**

**CountVectorizer without stemming CountVectorizer with stemming**

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |

**Tfidf Vectorizer without stemming Tfidf Vectorizer with stemming**

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |
| A close up of a sign  Description automatically generated | A close up of a sign  Description automatically generated |

The performance of the four models is similar. The stemmed version of CountVectorizer and Tfidf Vectorizer performed slightly better, increasing the accuracy of the model with 1%. Tfidf is the better vectorizer in general within this experiment. The stemmed version seems to classify more negative than positive reviews.

Next Support Vector Machine models is analyzed. Three different kernels are used: linear, rbf and poli. Different cost for each kernel were applied. The four data frames, generated from the four vectorizer were applied to SVM with 3 different kernels and 3 different cost values. An example of the first data frame from CounVectorizer without stemming is shown in the output bellow:

**SVM**

**CountVectorizer without stemming**

* **C = 50**

**Linear RGB POLI**

|  |  |  |
| --- | --- | --- |
| **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |
| **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |

**Acc 66% Acc 66% Acc 66%**

* **C = 0.0001**

|  |  |  |
| --- | --- | --- |
| **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |
| **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |

**Acc 71% Acc 65% Acc 50%**

* **C = 10,0000**

|  |  |  |
| --- | --- | --- |
| **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |
| **A screenshot of a cell phone  Description automatically generated** | **Screen of a cell phone  Description automatically generated** | **A screenshot of a cell phone  Description automatically generated** |

**Acc 61% Acc 63% Acc62%**

The same method is applied on the other 3 data frames.

**CountVectorizer with stemming**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear | RGB | POLI |
| C=50 | Acc 63% | Acc 65% | Acc 65% |
| C=0.0001 | Acc 71% | Acc 49% | Acc 49% |
| C=10,000 | Acc 59% | Acc 60% | Acc 61% |

**TfidfVectorizer without stemming**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear | RGB | POLI |
| C=50 | Acc 71% | Acc 71% | Acc 49% |
| C=0.0001 | Acc 70% | Acc 50% | Acc 50% |
| C=10,000 | Acc 52% | Acc 66% | Acc 68% |

**TfidfVectorizer with stemming**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Linear | RGB | POLI |
| C=50 | Acc 71% | Acc 71% | Acc 50% |
| C=0.0001 | Acc 69% | Acc 49% | Acc 49% |
| C=10,000 | Acc 60% | Acc 69% | Acc 69% |

It seems that the results are consistent with changing the cost. With c=50, the accuracies are the same in the different kernels, only ‘poli’ kernel showed really drastic change. With c= 0.0001, linear kernel seems to be the best option. With c= 10,000 the results for all ae really similar, probably ‘poli’ kernel performed best for most cases.

**Results**

MNB and SVM model were created to classify the sentiment of movie reviews. The best accuracy was received from a model with TfidfVectorizer, that added stemming (72%).

**Accuracy Table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **CountVect** | **CountVect STEM** | **TfidfVect** | **TfidfVect STEM** |
| **MNB** |  |  |  |  |  |
|  |  | **70%** | **71%** | **71%** | **72%** |
| **SVM** |  |  |  |  |  |
|  |  | |  |  |  | | --- | --- | --- | | **Lin** | **RGB** | **POLI** | | |  |  |  | | --- | --- | --- | | **Lin** | **RGB** | **POLI** | | |  |  |  | | --- | --- | --- | | **Lin** | **RGB** | **POLI** | | |  |  |  | | --- | --- | --- | | **Lin** | **RGB** | **POLI** | |
| **C=50** |  | |  |  |  | | --- | --- | --- | | **66%** | **66%** | **66%** | | |  |  |  | | --- | --- | --- | | **63%** | **65%** | **65%** | | |  |  |  | | --- | --- | --- | | **71%** | **71%** | **49%** | | |  |  |  | | --- | --- | --- | | **71%** | **71%** | **50%** | |
| **C=0.0001** |  | |  |  |  | | --- | --- | --- | | **71%** | **65%** | **50%** | | |  |  |  | | --- | --- | --- | | **71%** | **49%** | **49%** | | |  |  |  | | --- | --- | --- | | **70%** | **50%** | **50%** | | |  |  |  | | --- | --- | --- | | **69%** | **49%** | **49%** | |
| **C=10,000** |  | |  |  |  | | --- | --- | --- | | **61%** | **63%** | **62%** | | |  |  |  | | --- | --- | --- | | **59%** | **60%** | **61%** | | |  |  |  | | --- | --- | --- | | **52%** | **66%** | **68%** | | |  |  |  | | --- | --- | --- | | **60%** | **69%** | **69%** | |

Overall, the linear kernel classified best in each category. To get best fit model different cost values have to be experimented.

40 different models were created. Let us look **the most indicative words** for the ones with high accuracy.

**MNB**

**CountVect**

|  |  |
| --- | --- |
| **Important words in negative reviews:**  negative 2296.0 great  negative 1487.0 dont  negative 1409.0 funny  negative 1403.0 take  negative 1237.0 never  negative 1210.0 see  negative 1078.0 quite  negative 1050.0 lot  negative 1029.0 director  negative 988.0 life | **Important words in positive reviews:**  positive 1746.0 great  positive 1484.0 funny  positive 1315.0 get  positive 1310.0 quite  positive 1174.0 see  positive 1140.0 take  positive 1113.0 show  positive 1052.0 never  positive 934.0 lot  positive 908.0 dont |

**CountVect STEM**

|  |  |
| --- | --- |
| **Important words in negative reviews:**  negative 5850.0 know  negative 2510.0 give  negative 1642.0 hi  negative 1555.0 best  negative 1546.0 see  negative 1489.0 doe  negative 1488.0 role  negative 1430.0 show  negative 1424.0 look  negative 1386.0 find | **Important words in positive reviews:**  positive 4357.0 know  positive 2025.0 give  positive 1668.0 say  positive 1516.0 role  positive 1476.0 find  positive 1459.0 play  positive 1436.0 best  positive 1319.0 first  positive 1271.0 hi  positive 1182.0 seem |

**TfidfVect**

|  |  |
| --- | --- |
| **Important words in negative reviews:**  negative 240.89950591883428 great  negative 188.2556085676808 dont  negative 182.3743471314648 take  negative 176.09080302047462 funny  negative 162.3096804058115 never  negative 156.75460575717287 see  negative 145.06356698346605 quite  negative 143.84471194337192 director  negative 143.60740285671528 character  negative 142.32700238681596 life | **Important words in positive reviews:**  positive 221.4262880417276 get  positive 193.387036141983 great  positive 189.05295397590595 funny  positive 186.8628163257414 quite  positive 165.65542192519433 show  positive 155.571303368688 see  positive 152.08363820797712 know  positive 147.1108324400259 bad  positive 144.99309744455104 take  positive 142.46171905714502 never |

**TfidfVect STEM**

|  |  |
| --- | --- |
| **Important words in negative reviews:**  negative 453.09128077088246 know  negative 215.2968288231066 give  negative 171.633877094146 hi  negative 170.0427853728144 see  negative 166.07272990606674 best  negative 157.18877831273602 doe  negative 153.26292973216565 find  negative 151.2001540062235 show  negative 149.47480867664996 role  negative 148.90128234236064 look | **Important words in positive reviews:**  positive 369.24349036690086 know  positive 186.8601308938246 say  positive 182.47532146926363 first  positive 181.19236202227393 give  positive 168.73709265425512 play  positive 168.1833085855115 find  positive 163.57797185446793 guy  positive 160.35799492607052 role  positive 154.29335163384889 best  positive 146.86303436387348 see |

Another way to look at those indicative words is to plot them. Some SVM linear are shown below:

**SVM**

* Linear

CV not stem

A screenshot of a cell phone

Description automatically generated

Tfidf not stem

A screenshot of a cell phone

Description automatically generated

Different data set was used so Kaggle submission was not possible.

Classifying reviews based on sentiment alone proved to be a fairly easy exercise. There is so much labeled data to learn from, letters, words and sentences can be classified and looked from many different angles (look at the “valence” of a word (using an  external dictionary, something like Vader ), look at the words that follow  negation words and look at all the words spread out together in a sparse matrix  and let the computer find patterns for itself).

**Conclusion**

All people love to watch movies from time to time. Movies of various sorts like comedies, action, war, fiction, drama, and documentary. By watching movies , people get to relax from their daily activities. Movies allows humans to experience a life that they may never really get to experience by placing them on the shoes of the characters on the movie being watched.

If watching a movie showing a protagonist as a general during world war II, in some way experience through the movie the life of that general and his actions and adventures, even his problems and emotions become part of the audience while they are watching the movie. Real life story movies on the other hand, allows to witness the life of others which was experienced by them in real life at some point in time. People get to admire them on how they are able to solve and endure their real-life challenges. Science fiction is a different thing. It makes the audience imagine what would happen if things that are yet impossible today become possible in the future. Sci-fi movies are an attempt to predict the conditions of the future through a movie.

The simple truth is that probably 90% of the people read the reviews about a movie before they watch it on the big screen. Are the reviews important? Yes, for any type of business.

A vase of flowers on a table

Description automatically generated